

REMARKS

Prior to entry of this paper, Claims 1-11 and 13-29 were pending. Claims 1-11 and 13-29 were rejected. In this paper, Claims 6 and 25 are amended to create grammatical informalities; no claims are cancelled or added. Claims 1-11 and 13-29 are currently pending. No new matter is added by way of this amendment. For at least the following reasons, Applicants respectfully submit that each of the presently pending claims is now in condition for allowance.

Drawing & Specification Objections

The changes to Figure 2 are objected to because they introduce new matter, i.e. a memory and processor. The amendment filed on 06/07/2006 is objected to under 35 U.S.C. 132 (a) because it introduces new matter into the disclosure. Applicants traverse these rejections.

First, Applicants note that Figure 4, as confirmed by the Examiner, is directed to a software implementation. Applicants note that it is well known in the art that logic/actions are best implemented on a device with a processor and memory when the functionality of the logic requires relatively constant updates and rollback features. As disclosed by the original specification, the claimed network accelerator is responsible for the operation and functionality of such logic. For example, frequent updates are often necessary for logic that proxies network connections, as well as manage the operability of various IP protocols on a network, which may also require updates on a relatively frequent basis to keep up with changes to network connection standards and communication/IP protocols.

Although the disclosed network accelerator is clearly intended to be implemented through the use of a processor/memory arrangement, it is conceivable that some of its functionalities could be handled by custom hardware (firmware), which doesn't employ a processor or separate memory, e.g., Application-Specific Integrated Circuits (ASICs) and/or combinational logic. And in those other cases where a relatively finite set of non-changing functionalities are implemented by an invention, such firmware might be a preferable option for enablement. However, that is not the instant case. Rather, to maintain the operability of the network accelerator over a network,

relatively constant updates for its various functionalities are necessary to maintain communication, such as updates to the logic for proxying connections and the logic for maintaining the IP protocol stacks. Clearly, to update a firmware solution each time an internet/network standard and/or IP protocol changed would be time consuming, expensive and generally impractical. Thus, a processor/memory configuration that can quickly install and/or roll back logic updates to maintain the disclosed functionalities of the network accelerator is the most realistic, intuitive, and inherent solution.

For at least these reasons, the processor and memory added to Figure 2 and in the specification are inherent to one of skill in the art and are not new matter. The objection to the drawings and specification are moot and should be withdrawn.

Claim Rejections – 35 U.S.C. § 112

Claims 25-29 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. Regarding Claim 25, the Office Action alleges that the specification does not disclose the claimed memory or processor. However, for at least the reasons given above in relation to the objections to the drawings and specification, this rejection has been overcome and Claim 25 is now allowable. Claims 26-29 are allowable for at least the same reasons as Claim 25 upon which they depend. Applicants respectfully traverse this rejection and requests that this rejection be withdrawn.

Claim Rejections – 35 U.S.C. § 103

Claims 1, 2, 6-11, 13-17, 19-22, 24-27, and 29 are rejected under 35 U.S.C. 103 (a) as being unpatentable over Bartlett et al. (US 2003/0177396) in view of Hypertext Transfer Protocol HTTP/1.1 Standard, paragraphs 8.1, page 38. Claims 3-5, 18, 23, and 28 are rejected under 35 U.S.C. 103 (a) as being unpatentable over Bartlett et al. (US 2003/0177396) in view of Hypertext Transfer Protocol HTTP/1.1 Standard, paragraphs 8.1, page 38, and further in view of Dillon et al. (US 6,658,463). Applicants traverse these rejections.

Regarding Claim 1, Applicants hold that neither Bartlett nor the HTTP/1.1 Standard teach or suggest “striping the transmitting of process packet traffic”, as claimed, in part, by at least Claim 1. Applicants agree with the Examiner that Bartlett does not specify stripping data streams or packets over TCP connections that are parallel persistent connections. However, Applicants respectfully deny the motivation to combine with the HTTP/1.1 Standard.

Regarding striping, Bartlett states that the PEP path selection makes “sure that all of the packets related to the same traffic flow (e.g., the same TCP connection) *take the same path* (although it is also possible to send segments of the same TCP connection via different paths, this segment ‘splitting’ can have *negative side effects*).” (Emphasis added; See Bartlett, pg 10, paragraph 108). By making sure all the packets *take the same path*, Bartlett essentially rejects the notion of striping (or dividing) the transmitting of processed packet traffic. Consequently, Bartlett not only does not teach or suggest striping packets, Bartlett also generally teaches away from this concept by discussing negative side effects. Therefore, the HTTP/1.1 Standard cannot cure this defect since Bartlett teaches away from this additional limitation.

Further, the section of HTTP/1.1 Standard cited by the Examiner does not appear to teach or suggest striping. Instead, section 8.1 of the HTTP/1.1 Standard teaches *persistent* connections generally and the concept of “pipelining” or “sending multiple requests without waiting for each response.” See HTTP/1.1 Standard, page 39, paragraph 8.1.2.2. As defined by the specification, the “transmitted data stream is then *divided*, or “striped,” across the multiple parallel TCP connections.” See Specification, page 4, lines 26-27. Pipelining, or sending multiple requests without waiting for each response, is not the same as dividing a transmitted data stream across multiple parallel TCP connections. Therefore, section 8.1 of the HTTP/1.1 Standard does not appear to teach or suggest striping the transmitting of processed packet traffic. Therefore, even if these two prior art references are combined (which the Applicants deny), neither Bartlett nor the HTTP/1.1 Standard, either alone or in combination, teach or suggest striping the transmitting of process packet traffic. Thus, Claim 1 is not obvious in view of Bartlett and the HTTP/1.1 Standard. Applicants respectfully request that this rejection be withdrawn.

